DOCUMENT RESUME

ED 048 747 EM 008 804

AUTHOR	Parker, Edwin B.
TITLE	Requirements for SPIRES II. An External Specification for the Stanford Public Information Retrieval System.
INSTITUTION	Stanford Univ., Calif. Computation Center.
SPONS AGENCY	National Science Foundation, Washington, D.C.
PUB DATE	Apr 71
NOTE	58p.
EDRS PRICE	EDRS Price MF-$0.65 HC-$3.29
DESCRIPTORS	*Computer Oriented Programs, Computer Programs, Data Analysis, Guides, Information Retrieval, Information Storage, *Information Systems, *Man Machine Systems, *Manuals, Research Tools, Time Sharing
IDENTIFIERS	SPIRES, *Stanford Public Information Retrieval System

ABSTRACT

SPIRES (Stanford Public Information Retrieval System) is a computerized information storage and retrieval system intended for use by students and faculty members who have little knowledge of computers but who need rapid and sophisticated retrieval and analysis. The functions and capabilities of the system from the user's point of view are described in this manual. The three categories of users who may interact with the system are defined as searcher, updater, and file manager. The role of the user consultant is described as managing the system and providing information to facilitate use. Such functional areas as file definition, file maintenance, searching, retrieval, and output are surveyed. The hardware and software configuration of the system and the effect of this configuration on data organization is described. A transcription of a file definition session illustrates the method of file definition used by the system. The process used to maintain and update files and the commands used in searching and retrieval are listed and explained. Various options for displaying output are described. (JY)

ED048747

U.S. DEPARTMENT OF HEALTH, EDUCATION
& WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

REQUIREMENTS
FOR
SPIRES II

AN EXTERNAL SPECIFICATION FOR THE STANFORD PUBLIC INFORMATION RETRIEVAL SYSTEM

SPIRES
a project of
the National Science Foundation
Edwin B. Parker, Principal Investigator

APRIL 1971

SPIRES/BALLOTS PROJECT
STANFORD UNIVERSITY
STANFORD, CALIFORNIA
94305

EM008 804

A NOTE TO THE READER

This report describes a new kind of information system at Stanford University--one that provides both information and a means of selecting and analyzing it. In 1972, SPIRES, a computer information retrieval system, will be available for faculty and students to use in conducting research. We are distributing this report widely in order to get comments on the functional description of the system. We would like to know, for example:

Is it clear what you can do with each of the SPIRES commands?

Are there things you would like to do that are not in SPIRES as it is described in this report?

Your comments should be sent to Professor Edwin B. Parker, Cypress Hall, Stanford University, Stanford, California 94305

WHAT IS SPIRES?

SPIRES (Stanford Public Information REtrieval System) is a computer information storage and retrieval system being developed at Stanford with assistance from the National Science Foundation. SPIRES offers:

-Ease of use.
-A variety of services.
-Reasonable cost.
-Daily availability.

WHO WILL USE SPIRES?

SPIRES is intended for the faculty member or student who has little knowledge of computers but who needs rapid and sophisticated retrieval and analysis of information.

WHY IS SPIRES NEEDED?

Even greater amounts of information (such as survey data and bibliographic references) are being made available from government sources, from professional associations, and from private organizations. Examples of these are census tapes, tapes from the American Psychological Association, and Chemical Abstracts research reports. More data is being produced at Stanford. All such data can be entered into SPIRES and used for faculty and student research.

WHAT CAN SPIRES DO?

With SPIRES you can create a collection of information (called a file) and search or analyze it. If you want to run successive searches or analyses based on previous results, you may use SPIRES interactively (on-line) at a computer terminal. Terminals are available at the computer center; they also can be rented and located almost anywhere in the campus community. The results of your work with the file can be printed out at your terminal or on a high-speed printer at the computer center in any format you choose. These are only a few of the things that can be done with SPIRES. A complete description of the system's capabilities is in this report.

CONTENTS

		Page
	ACKNOWLEDGMENTS	1
	INTRODUCTION	2
Chapter		
1.0	THE SPIRES USER	
	1.1 User Audience	1-1
	1.2 The User Consultant	1-1
	1.3 General Comments	1-2
2.0	THE SPIRES SYSTEM	
	2.1 File Definition	2-1
	2.2 File Maintenance and Update	2-1
	2.3 Search and Retrieval	2-1
	2.4 Display	2-1
3.0	SOFTWARE, FACILITIES, AND CONCEPTS	
	3.1 The SPIRES Environment	3-1
	3.1.1 MILTEN	3-1
	3.1.2 WYLBUR	3-1
	3.1.3 STSM	3-1
	3.2 The SPIRES Facilities	3-3
	3.2.1 The 2741 Terminal	3-3
	3.2.2 The CRT Terminal	3-4
	3.2.3 SPIRES System Commands	3-5
	3.3 STSM System Commands	3-8
	3.4 Definitions and Concepts	3-8
	3.4.1 Data Elements	3-8
	3.4.2 Records	3-9
	3.4.3 Indexes	3-13
	3.4.4 Files	3-14
4.0	FILE DEFINITION	
	4.1 Design	4-1
	4.2 A Typical Session	4-1
5.0	FILE MAINTENANCE AND UPDATE	
	5.1 Concepts and Definitions	5-1
	5.1.1 General	5-1
	5.1.2 External Data Format	5-1
	5.2 On-Line Entry and Deferred Update ..	5-3
	5.3 On-Line File Management Commands ..	5-3
	5.3.1 Update Execution Commands ..	5-3
	5.3.2 Record and Data Element Commands	5-4

5.4 Batch Services 5-7
5.4.1 Batch Update 5-7
5.4.2 Batch Conversions 5-8
5.4.3 Batch Utilities 5-8

6.0 SEARCHING AND RETRIEVING

6.1 General 6-1
6.2 Search Language 6-2
6.2.1 Browse Commands 6-2
6.2.2 Retrieval Commands 6-4
6.2.3 Batch Searching 6-10

7.0 DISPLAYING OUTPUT

7.1 Output Formats 7-1
7.1.1 System Default Format 7-1
7.1.2 Formats Defined at File Definition Time 7-1
7.1.3 Formats Defined During Session Time 7-1
7.2 Output Language 7-2
7.2.1 Type Command 7-2
7.2.2 Output Command 7-3
7.2.3 CRT Commands 7-4

ACKNOWLEDGMENTS

The activity reported in this document was supported by the National Science Foundation under grants GN 830 and GN 830.1.

This document was jointly authored by members of the Technical Development Group, Project SPIRES/BALLOTS:

Douglas Ferguson
Richard Guertin
William Kiefer
Herbert Ludwig
Thomas Martin
James Marsheck
John Schroeder
Cheryl Stevens
Jerrold West

The ACTION Analyzer must be singled out as an accomplishment of particular note. The ACTION BNF metalanguage is a powerful and concise tool for on-line command language definition; the Analyser has been invaluable in pointing out problem areas in early versions of the language. It will be of further service in the design of the system and during implementation of the parser.

INTRODUCTION

Purpose

This document was created as an external specification for SPIRES II (Stanford Public Information REtrieval System). External specifications describe system functions and capabilities from the user's viewpoint. This document is therefore cast in the form of a User's Reference Manual.

It should have three uses. First, as a basis for system development. Second, as a prospectus for users, present and future. (It is hoped that readers of the document will take the opportunity to criticize its contents and to offer suggestions toward improving the usefulness and convenience of the command language.) Third, as a basis for a future User's Manual that will contain many more examples and explanations and will be organized for easy reference.

The user command language documented here is a first formulation; it will very likely be modified in light of user experience. The language design and implementation completed to date has taken this into account: changes in the language, in general, will require little effort, and routines for gathering feedback information on user behavior will be part of the completed system.

Organization

Chapter 1 briefly describes potential users, dividing them into three categories. It also describes the functions of the user consultant. The chapter ends with a note on the way in which the command language is presented in following chapters.

Chapter 2 is a brief overview of the SPIRES system. It surveys the functional areas in which a user may operate: file definition, file maintenance, searching and retrieval, and output. (These are considered as separate areas solely for descriptive purposes. The user need not tell the system the area in which he wishes to work.)

Chapter 3 describes the general setting in which SPIRES will operate, including interfacing system software (WYLBUR, the Stanford Time-Sharing Monitor, and MILTEN). SPIRES terminals are also described, as well as SPIRES commands. WYLBUR, STSM, and MILTEN commands are touched on. The chapter finishes with a detailed discussion of the basic concepts of data organization in SPIRES.

Chapter 4 deals with file definition. Following an introductory section, the transcription of a file definition session illustrates the process.

Chapter 5 explains the file maintenance and update function. Examples of various possible operations are given, as well as detailed instructions on update commands.

Chapter 6 explains searching and retrieval. It lists the commands and their uses, and gives examples for each.

Chapter 7 discusses output formats, how and when the user can define them, and how the user can command output.

****

CHAPTER 1: THE SPIRES USER

1.1 User Audience

There will be three kinds of user. The category or categories to which a user belongs depends on how he interacts with the computer system. The categories are:

-Searcher
-Updater
-File Manager

The searcher submits search criteria via his computer terminal; the system uses these to locate data of interest to him. The searcher can enter more criteria, interactively, and either narrow or widen the scope of his search.

The updater uses one of several possible input media to submit new information for inclusion, delete obsolete information, or change existing information in a file.

The file manager is responsible for maintaining the contents of a particular file and for allowing or not allowing certain persons to search or update the file. He also defines the file, and subsequently supervises updating.

For small, simple files, one person may fill all of these categories. For larger, more complex files or files of interest to many people, a division of labor is likely to lead to more specialization, with one person serving as file manager, others updating the file, and still others using the file only as searchers.

1.2 The User Consultant

Assisting the user in all these activities will be the user consultant. If a person wishes to use SPIRES he may contact the user consultant--

John Jones
Polya 000
321-2300, x0000

--who will provide information about:

(1) How to open an account at the Campus Facility in order to use the SPIRES system.

(2) How to set up the user's file.

(3) Security restrictions the user may want to place on his file (e.g., who has access, update privileges).

(4) Terminal installation charges and other charges relating to operation under SPIRES and the Campus Facility.

(5) Tutorial classes given by the User's Services Group that describe the SPIRES concepts and facilities.

The user consultant determines whether or not a particular information storage and retrieval application can be run under SPIRES. He has full operational control over the total system and any problems encountered in its use or operation should be discussed with him.

The user consultant exists to help those who need help. If the user already has an account number set up under the Campus Facility, he may choose to build files using SPIRES without going through the user consultant. If the user has an inordinate number of records he wishes to input, he should first obtain information from the user consultant on the maximum number of records allowable. This number may vary depending on the particular application.

1.3 General Comments

In the sections that follow, the command language is presented with all verbs in their long form. Although not shown in this document, any command verb may be abbreviated to its first three characters when input. Furthermore, the user need not type commands strictly in upper or lower case; he may use either or a mixture. For clarity, commands are shown here with command verbs and system prompts in upper case and responses and parameters in lower case. Descriptions of commands are frequently followed by an *example* of the command as it is input by the user and the *response* displayed by the system.

****

CHAPTER 2: THE SPIRES SYSTEM

2.1 File Definition

File definition permits the user (the file manager) to completely define the structure of his file (see chapter 4). The file can be very simple or very complex. SPIRES allows the file manager to define hierarchical structures (ten deep) for his file. The file definition language gives him the power to define a number of different types of records to be included in his file. For each type of record that he defines, the system will maintain a data set that acts as an index to the record values.

The terminology the file manager uses in defining the empty file structure is later used in file maintenance and retrieval.

2.2 File Maintenance and Update

After the file manager has defined his file, he must have the means to maintain it. Maintenance facilities are provided by the File Management Command Language (see chapter 5). In the maintenance mode one can add, delete, replace, and modify existing records and/or data elements. Using the WYLBUR text editor as a complementary system gives the file manager a comprehensive set of update commands.

2.3 Search and Retrieval

The retrieval functions allow the user to search his file interactively. He can also obtain information about his file. Using the SPIRES interface with the Stanford Time-Sharing Monitor (STSM), he can perform statistical and computational analysis, and can generate reports based on selected records in the file. (See chapter 6.)

2.4 Display

The output language (see chapter 7) allows the user to display record or field information in various formats. The user can specify and define certain output formats to be used frequently. He can also select the device and method for displaying his information. He can:

(1) Print out information at the terminal.

(2) Print out information at the high-speed printer.

(3) Print out information in the user's WYLBUR DATA SET.

(4) Display information on a CRT (cathode ray tube) display device.

****

CHAPTER 3:
SOFTWARE, FACILITIES, AND CONCEPTS

3.1 The SPIRES Environment

SPIRES is a re-entrant subprocessor in the STSM partition of the IBM 360/67 computer system at the Campus Facility of the Stanford Computation Center. Figure 1 shows the relationship of SPIRES to its environment. Note that SPIRES interfaces with the WYLBUR text editor, thus allowing users to apply WYLBUR's text-editing commands to data sets generated from SPIRES search strategies.

3.1.1 MILTEN

MILTEN is the communications subsystem serving the IBM 360/67 computer. It handles up to 88 typewriter terminals concurrently through a 2703 Communication Adapter. MILTEN is now being modified to handle CRT devices as well, using a PDP-11 front-end computer as an interface. MILTEN switches terminal messages between the user and WYLBUR or STSM as appropriate.

3.1.2 WYLBUR

WYLBUR is a text editor with which one can collect textual data into lines; these can in turn be copied, deleted, inserted, moved, or replaced. One can also do intraline editing--replacing, deleting, or inserting strings of characters. Editing commands operate on ranges of lines, set by the user and specified by either line number or line content.

WYLBUR communicates with the IBM 360/67 operating (O/S) system to permit using, saving, deleting, or displaying O/S data sets. Communication with RALPH, the system job stream processor, permits remote job entry into O/S batch, retrieval of printed output from O/S batch jobs, and queries about the status of O/S batch jobs. Finally, communication between WYLBUR and STSM permits all of WYLBUR's facilities to be used as extensions of STSM's supervisor services.

3.1.3 STSM

STSM is a time-sharing submonitor that operates under the OS/360 system in conjunction with WYLBUR. STSM permits user programs to be executed in a time-shared mode. It appears to the user, in most cases, as an extension of the WYLBUR commands plus a set of programming conventions. STSM provides the Campus Facility with:

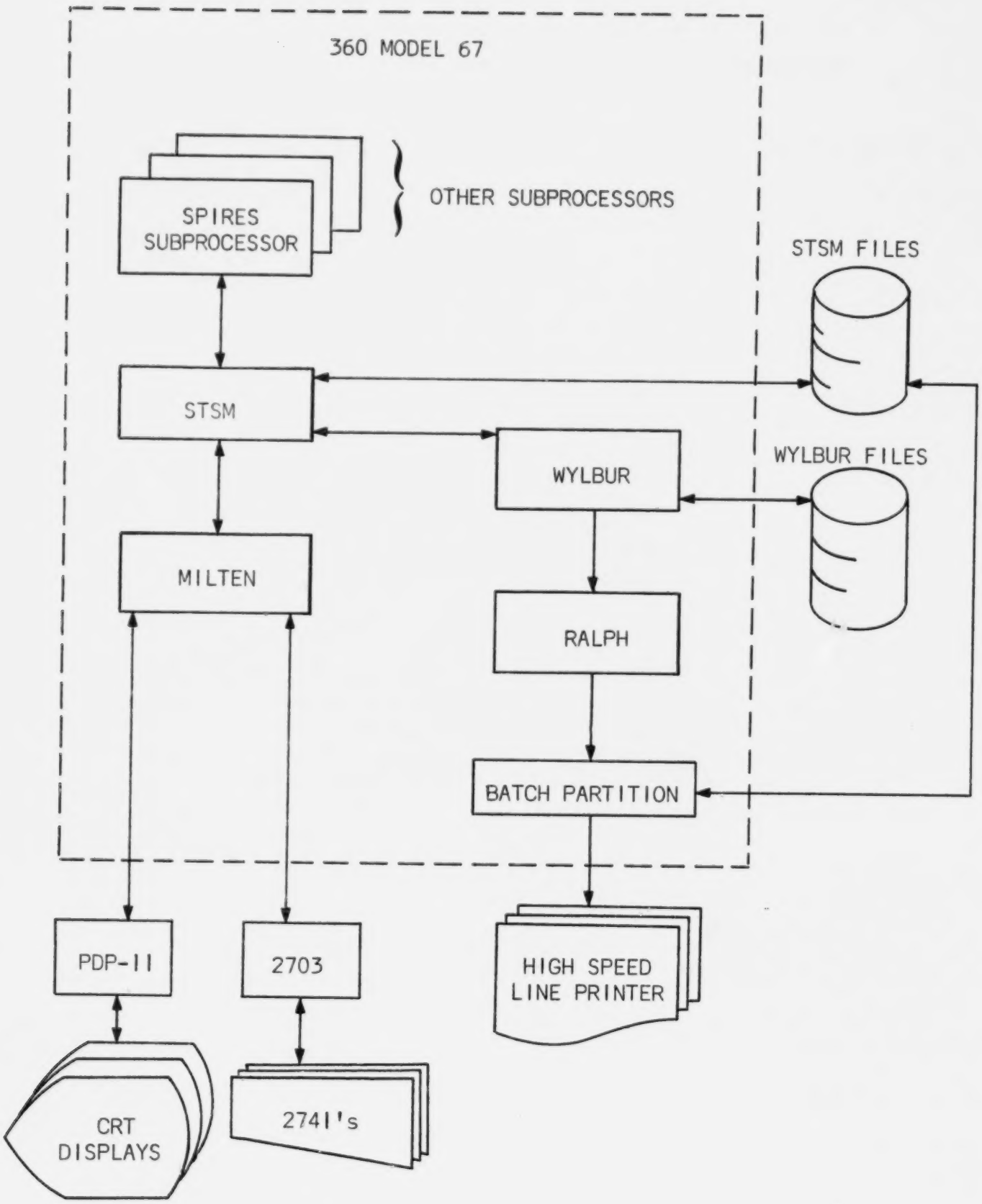


FIGURE 1: The SPIRES Environment

(1) A set of system subprocessors that can be used by many users simultaneously. Such subprocessors (BASIC, FORTRAN, and SPIRES among them) operate as normal user programs, except a single copy serves all users.

(2) The ability to load and execute any user program.

(3) Aids to debugging user programs.

(4) A file system on the 2314 disk for holding system and user data.

3.2 The SPIRES Facilities

3.2.1 The 2741 Terminal:

3.2.1.1 Device Description

The IBM 2741 Typewriter Terminal is similar in appearance to a selectric typewriter. The difference is, of course, that it is hooked to a computer via a leased or dialup telephone line, and hence can transmit the user's input to the computer, and print back the computer's answer to the user.

Figure 2 shows the location of data and function keys on the 2741 terminal keyboard.

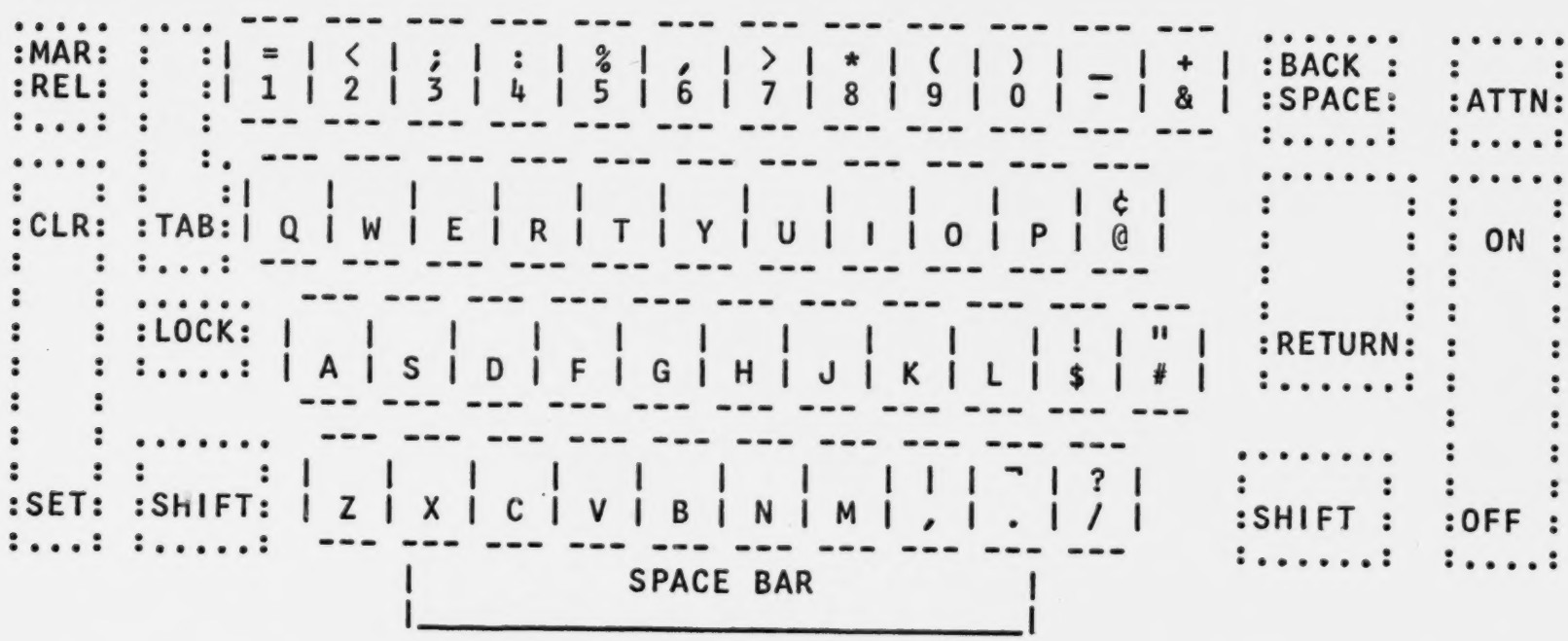


FIGURE 2: The IBM 2741 Keyboard

3.2.1.2 Function Keys

Function keys do not transmit data characters to the computer. Instead, they transmit one or more control characters, which have some special meaning depending on what the user is doing at the time.

(1) RETURN - This key causes the typing ball to return to the left-hand margin, and signals to the computer that the line's end has been reached.

(2) ATTN - This key tells the computer that it should interrupt the current activity and take appropriate action, such as prompting for a new user command.

(3) BACKSPACE - This key moves the typing ball one character to the left. Any typed character that is backspaced over is deleted. Note: A command that permits backspacing to overstrike characters (e.g underlining) is also available to users.

(4) ON-OFF - This switch controls the power to the terminal. It should not be altered during a terminal session.

3.2.2 The CRT Terminal

At present, several CRT devices are being evaluated for use as SPIRES terminals. The general characteristics of the terminal will be:

-9,600-Baud transmission rate.
-Upper/lower case.
-1,000-character screen.
-Character/line edit capability.
-Paging and scrolling from a buffer of at least 2,000 characters.
-Proven reliability.
-Cost within acceptable limits.
-Hard copy capability as an add-on.

As stated above, CRT terminals will communicate with the 360/67 via a PDP-11 front-end computer.

When a final choice is made, a set of extensions to the SPIRES command language will be implemented to make use of the

functional capabilities of the CRT. These extensions will be available to the SPIRES II users no later than six months after initial system installation.

3.2.3 SPIRES SYSTEM COMMANDS:

3.2.3.1 The EXPLAIN Command

EXPLAIN <Command Verb>

This command causes an explanation of the appropriate verb to be displayed at the terminal. If no command verb is indicated, then a list of command verbs is displayed in alpha sequence.

Example: EXPLAIN FIND

Response: The command FIND enables the user to search all records in his file to find those with a particular alpha value or to find a character string in a data element. If he wants to the user may join several values using the logical operators "and," "or," or "not."

3.2.3.2 The EXAMPLE Command

EXAMPLE <Command Verb>

This command causes an example of the use of the verb in question to be displayed at the terminal.

Example: EXAMPLE FIND

Response: FIND Name Smith and Title Aardvark

3.2.3.3 The DISPLAY File Command

This command allows the user to display certain parameters associated with the files that he is privileged to use. The command has two possible forms:

(1) DISPLAY (PUBLIC or PRIVATE) FILES

(2) DISPLAY FILE <filename>

If the option PUBLIC or PRIVATE is not included, the user will receive a listing of all the public and private files that he is allowed access to under his present account number. If the PUBLIC option is designated, the user receives a list of all the public files that he is allowed access to; if the PRIVATE option is designated, he receives a list of all the private files that he is allowed access to. The second display command refers

only to one particular file and lists for the user some pertinent parameters associated with that file. Anyone who has access to a terminal may retrieve from a public file. Private files are those maintained by individuals or organizations for their own use. Present system design constraints do not allow a user to build and use a file on a removable disk pack. This constraint will be removed at a future date. The number of people using the file is primarily of concern to the manager of the file, although the general user may be interested.

Example: DISPLAY FILES

Response:

FILES	PUB	PRIV	SIZE	USERS
marc	x	-	25K	2
preprint	x	-	5K	3
ornithology	-	x	1K	1
jones	-	x	3K	-
desy	x	-	28K	0
ipf	-	x	50K	3

3.2.3.4 The SELECT (RESULT) <filename/resultname> Command

Once a user has determined the name of the file he wishes to retrieve from, he can enter a SELECT request. If the file is one to which access is limited, the system will ask for a password (security key) that the user must enter correctly to be allowed access to the information in the file. The SELECT RESULT <resultname> command is utilized in conjunction with the STORE RESULT command that allows the user to save the results from a particular search strategy just executed on his file (see 6.2.2.4).

(In all following examples the user command appears after the "-?")

Examples:

a. -? SELECT preprints (search a public file)

b. -? SELECT ipf (search a public file)

c. -? SELECT greekhist (search a private file)
SECURITY KEY?
-? AXB (overstruck to prevent readability)
-? FIND AUTHOR ronson or TITLE athenian

d. -? SELECT RESULT thisfile
(search a file beginning with results saved earlier in the session)

3.2.3.5 The DISPLAY (INDEXES/ELEMENTS/FORMATS/CHARACTERISTICS) Command

This command allows the user to request that various properties of the file he has chosen be listed at his terminal.

The request "DISPLAY INDEXES" will obtain a list of the names of the indexes that exist for the file. The request "DISPLAY ELEMENTS" will obtain a list of the data elements defined for the file. The request "DISPLAY FORMATS" will obtain a display of the standard formats for the file. The name of each format will be followed by a list of the elements that will be displayed when that format is specified for the output of records found by a search. The request "DISPLAY CHARACTERISTICS" will obtain a list of all the properties of the file mentioned above. In addition, the prose description of the file that was entered by the file manager at its creation will be given. In general, items displayed will be those to which the requester is allowed access. If, for example, the user requests "DISPLAY ELEMENTS," he will see only these element names that he is permitted to used. (See 5.3.2.5.)

Example: DISPLAY INDEXES

Response:
AUTHOR
TITLE
TOPIC
REPORT NUMBER

Example: DISPLAY ELEMENTS

Response:
JOURNAL
AUTHOR
TITLE
AFFILIATION
DATE
LANGUAGE
NOTES

Example: DISPLAY FORMATS

Response:
NORMAL
1. TITLE
2. AUTHOR
3. DATE
BIBLIO
1. AUTHOR
2. TITLE
3. JOURNAL,VOLUME,PAGE
4. DATE

3.2.3.6 The HELP Command

If the user leaves his terminal for a few moments and cannot remember where he was in a series of commands, he can type "HELP," and SPIRES will display the last command he executed. For the inexperienced user, SPIRES will indicate what options are available and point him toward the EXPLAIN and EXAMPLE commands.

3.2.3.7 The TO SPIRES Command

The user may wish to send a message to SPIRES to bring to the attention of the user consultant some peculiarity he has discovered or some problem he may be having with the system. This can be done with the TO SPIRES command. The message is logged, and the user consultant reads it later during the day or the next morning.

Example: TO SPIRES: Having trouble, EXTRACT command, Smith

3.3 STSM System Commands

The system commands available under MILTEN, WYLBUR, and STSM such as SET TERSE, SHOW TIME, SHOW FILES, etc., are explained in detail in the WYLBUR and STSM Reference Manuals. These manuals can be obtained from the Accounting Information and Documentation Services (AIDS) Office of the Stanford Computation Center.

3.4 Definitions and Concepts

The heart of any information storage and retrieval system is its data files. An understanding of the basic concepts of data organization in SPIRES should aid the user whether he is a searcher, an updater, or a file manager.

3.4.1 Data Elements

The basic data-file building blocks are data elements. These are primary units of information, such as "AUTHOR," "SUBJECT HEADING," "TITLE," etc. Data elements are not to be confused with values of data elements.

DATA ELEMENT	VALUE
Author	Newton, Isaac, Sir
Subject	Particle Physics
Title	Muon-pion interaction
Age	58
Status	Dead
SS (Social Sec. #)	556-67-9384

Values may be thought of as specific occurrences of a data element.

Data elements may occur in groups, and may form hierarchies. A simple example of a hierarchy is "date."

date
   month
   day
   year.

"Month," "day," and "year" are hierarchically subordinate to "date." Note that subordination is denoted by indentation; that style will be used hereafter wherever hierarchies are shown. A more complex example is found in patient medical files:

Patient Name				
Address				
Visits:	1	2	3	...
Date	1/2/69	2/8/69	6/1/69	...
Diagnosis	Heart C.	ok	ok	...
Treatment	Operation	--	--	...

There are also a number of restrictions that may be placed on a data element.

(1) It must occur a certain number of times.
(2) It may not occur more than a certain number of times.
(3) A value must satisfy certain validity tests:
    (a) It must be a least 'x' characters long.
    (b) It cannot be more than 'y' characters long.
    (c) It must be numeric, alphabetic, or some other set of characters.
    (d) It must be a code.
    (e) It must satisfy some special restriction, e.g., date.
(4) Data elements must be sorted into some special order, e.g., alphabetic.
(5) Data elements must be converted before storage, e.g., to fixed binary, real number, pointer.

3.4.2 Records

Records are merely groups of logically related data elements. Records can refer to other records by using connectors or pointers. For instance, a record containing the author value "SMITH" could point to a record containing a bibliographic description of a book by SMITH. If several such pointer records are grouped and organized to facilitate easy location of a particular data element value, then that group of records is termed an INDEX.

3.4.2.1 Parameter Processing

We do not want to limit ourselves to discussing only bibliographic applications. The SPIRES system has been designed

so that other types of retrieval tasks, such as "parameter processing," can be handled. Parameter processing permits the user to search all the data elements in a record without requiring an index for each element. Logically connected interrogations are usually applied to the file to obtain the desired results. (An arbitrary limitation on file size will be imposed on the user for this kind of searching.) For example, a search strategy may be determined that tests many data elements in a file (containing medical, sociological, or demographic data) for a particular value before arriving at a specific set (a subset) of records that satisfy the search criteria. If the data elements of a record are numerous, it might be expensive to index all of them (indexing is discussed in the next section); the SPIRES system allows the user to define his file in such a manner that parameter processing can easily be done instead.

3.4.2.2 Coding Fields

In most cases of parameter processing, the data elements will include a code describing some parameter's existence.

Example: 9(2) Prostatism:

Dribbling	A
Obstruction	B
Hesitancy	C
Nocturia	D
Dysuria	E
Frequency	F
Decreased Stream	G
Urgency	H
None	N

In this example the condition of prostatism is given a code (A-H, N) that describes various symptoms. The user must remember that if codes are used for data element values, then those same codes must be used when the user is searching.

Example: 11 Status - Pre Rx (treatment)

(1)	Castration	Y	N
(2)	Estrogen (Start)	Y	N
(3)	Estrogen (Stop)	Y	N

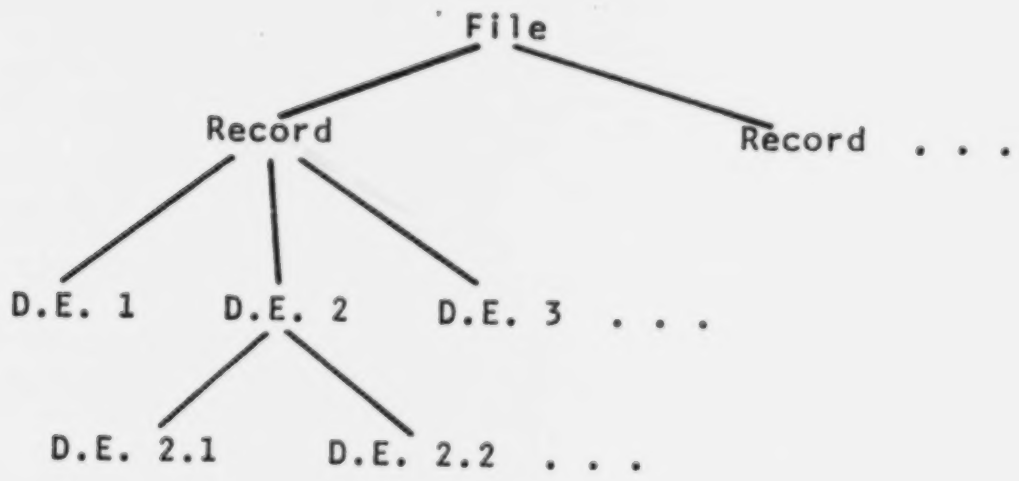
The first example permits any of the combinations of A-H, whereas the second permits only a single entry, Y or N. Thus, the code "ABCDEFGH" might be encountered (very unlikely). In the second case, the information might, at first glance, be thought of as binary information (0 for N, 1 for Y). But allowance must be made for an entry that specifies "no information" or "cannot be determined." Thus, instead of storing this information as a bit, one should store it as a byte (an 8-bit code); it is then similar to the first example, the only difference being the

number of bytes reserved for each data element. If the maximum number of codes used by a data element can be determined, then the field can be fixed in length. In some cases, this method of storing the data will waste space, but the amount wasted should be minimal. Note: Parameter-processing applications may be applied only to files or to subsets of files not exceeding some arbitrary length. This length must be determined with the user consultant.

Within a SPIRES record, every property is identified as an individual field: a patient's name in a hospital record, an address, an invoice number, etc. Once record structure is determined by the user, fields are declared, named, and identified to SPIRES during file definition. They are then available for any type of retrieval from the file. The fields of a record can be numeric integer such as "Age," numeric real such as "purchases" on a charge account (XX.XX), alphabetic such as "Name" or "Address"; they can also be dates, which can take any of the common forms.

3.4.2.3 Data Elements

A record consists of data elements. Certain data elements within a record can form structures. SPIRES provides the capability for nested structures, no more than ten deep.



SPIRES provides the user with the opportunity to specify completely his own file organization or structure. Thus, the user does not have to be concerned about using a fixed field, a fixed word length, or a fixed format. He is not bound by a set of rigid rules on record size, length, etc. Instead, he may first compile a working list of all the data elements to be included in a record, specifying whether or not a field is singular or multiple (has subfields). Example: Suppose that we want to create a SPIRES file of heart-transplant

patients; we have determined that we want to include the following information (data elements) in a patient's record:

DATA ELEMENT NAME	SINGLE/MULTIPLE
Name of Patient	S
Hospital Record Number	S
Home Address	S
Date of Birth	S
Home Phone	S
Diagnosis	M
Cardiac Diagnosis	
Vascular Diagnosis	
Admission Date	M
Year	
Month	
Day	
Date of Operation	
Status at Time of Dischg.	
Date of Death	S

A typical patient record would have the structure:

Name	Record #	Address	Birthdate	Home Phone
Homer R. Jones	237863	28 Sixth Ave. Torrance, CA.	19100410	258-6789

Diagnosis

	1.	2.	3.	....
Cardiac Diag.	(text)	(text)	(text)	....
Vascular Diag.	(text)	(text)	(text)	....
Admission Date	19681120	19690218	19690307	....
Operation Date	----	19690210	19690309	....
Status	(text)	(text)	(text)	....

Date of Death

Note the structure of the data element "Diagnosis." An unlimited number of entries can be associated with this data element, each denoting a separate visit.

The user must also determine the type of each data element field that he includes as part of a record. For example, Name of Patient would be an alphabetic character string, whereas Hospital Record Number would be an integer; Date of Birth would be, obviously, a date--the same type of field as Admission Date and Operation Date. In the last two cases a distinction is made to indicate that these two fields are multiple.

After determining the type of each field and whether or not that field is singular or multiple, one can number and name the fields. Adding the restrictions that the user may specify, (see 3.4.1), one obtains the following:

FIELD	NAME	DESCRIPTION	RESTRICTIONS (1)	(2)	(3)	(4)	(5)
1	Name	Name of Patient	x	x			
2	Number	Hospital Record Number	x	x	x		
3	Address	Home Address	x	x			
4	BirthDate	Date of Birth	x	x	x		
5	Phone	Home Phone	x	x			
6	Diagnosis			x			
6.1	CDiagnosis	Cardiac Diagnosis		x			
6.2	VDiagnosis	Vascular Diagnosis	x	x	x		
6.3	AdmDate	Admission Date		x		x	x
6.3.1	Yr	Year		x		x	x
6.3.2	M	Month		x		x	x
6.3.3	D	Day	x	x			
6.4	OpDate	Operation Date	x	x	x		
6.5	Status	Status at Discharge	x	x			
7	DDate	Date of Death	x	x	x		

The user has, in this example, taken the first step in defining the structure of his file. Note that there are two hierarchies apparent, one subordinate to the other: Diagnosis and Admission Date.

3.4.3 Indexes

If it is necessary to examine a collection of data records for all entries by "SMITH," an index containing author names and pointers to data records can be searched, and references to all data records with SMITH as the author quickly gathered. There are various kinds of indexes. A few types commonly used in bibliographic files are listed below.

AUTHOR: Author indexes contain personal names stored in some canonical form such as SNOW, C.P., LORD, III. Such indexes must be searchable through incomplete keys, such as SNOW or C. SNOW.

TITLE: Title indexes are usually composed of individual non-trivial title words taken from the values of the data element 'TITLE' in the data record collection. In order to build a title index consisting of non-trivial title words, it is first necessary to compare the title word with an exclusion list. This list contains such words as "and," "or," "not," "an," "if," etc.

TOPIC: Topic indexes are usually quite simple, even when each word in the topic is indexed, as in a title-word index. The values of the data element TOPIC within all records in the collection are placed "as is" in a topic index, with accompanying references to the data records from which each value came. Alternatively, topics may be indexed as an entire word string, with multiple blanks, wherever they occur, reduced to one blank.

3.4.4 Files

A file is simply all the records in a particular collection: the data records, the index records, the exclusion lists, and the stored descriptions of each different record format in the collection.

****

CHAPTER 4: FILE DEFINITION

4.1 Design

SPIRES' design allows for the building and searching of many different types of files. But before a file can be built, it has to be defined. Through file definition, the manager of a file specifies the file's structure and future uses. As the file grows, its structure may have to change to accommodate changing use and/or data. Since redefinition occasionally involves rebuilding the file, the original definition should be as complete and accurate as possible.

The file definition processor is like an automated consultant. Since most users will have had little experience in this process, it will explain why a question has to be asked, indicate the alternatives, recommend one of the alternatives, and then formally ask the question. Depending on the response, it will select additional questions to ask. Users who do not need this much assistance can skip the instruction and have the questions asked at once. Users who know of another file very similar to their own can use the other file's definition as a model. Users who wish to redefine their own files must use their previous definitions as models to be modified.

The user consultant is responsible for maintaining and supplementing the definition processor. Users should consult with him before beginning the formal process. He can advise them of the types of information they need, whether or not they might be better off combining forces with other users, and how to activate the definition processor.

In the remainder of this chapter, a sample session with the processor will be presented. The sample is merely illustrative and not to be taken as a strict guide. The definition processor is likely to be changed considerably as experience is acquired with SPIRES functioning and file requirements.

4.2 A Typical Session

Welcome to the SPIRES file definition processor. Do you intend to DEFINE or REDEFINE a file?

MODE? define (carriage return)

An experienced user would respond with a third alternative like "define unaided."

What would you like to call your file?

FILE NAME? donor (cr)

Although your file may be very similar to some other file, it is to your advantage to think through your requirements from scratch. However, if you really want to, you can use another file's definition as a model. If you do not want to use another file's definition, hit the carriage return.

FILE DEFINITION MODEL? (cr)

During this session you will be asked questions that will help you define the file you have called "donor." Only when the definition is complete and accurate can the file be created. The session may last more than two hours, depending on the size and complexity of the file being created.

The purpose of SPIRES is to allow individuals or groups of individuals to build and search data collections--files. A file may consist of more than one type of record. At the present time, primary records must follow one record type. Any other record types must act as indexes to occurrences of primary records. Other file managers have called their primary record type a DOCUMENT, CASE, INVOICE, or RECORD.

What would you like to call your record?

PRIMARY RECORD NAME? case (cr)

How many cases do you plan to put into your file the first time you enter cases?

INITIAL SIZE? 100 (cr)

Very often file use changes with size. In the initial stages there is growth with little searching. After some critical point, the file becomes useful for searching. In this case it is wise to postpone index building until the file reaches its critical size. Then the file can be redefined.

From now until the next redefinition will your file be used primarily for GROWTH, FINDING, or BOTH?

FILE USE? growth (cr)

Even though file "donor" will be used primarily for growth, you may wish to establish one or more search classes. Search classes set limits on the types of elements that may occur in a search request. One class of searchers might be able to specify only case identification numbers; another class of searchers

might be able to specify only some of the elements; a third class might be able to specify any of the elements. If you intend to establish search classes, give each a name and list them (separated by commas). If no search classes are specified, search requests can specify only the case identification number (see below, the paragraph following "OUTPUT LANGUAGE?").

FIND CLASSES? (cr)

Would you like to limit the individuals who can search your file? If so, list their account numbers (separated by commas). Otherwise your file will be public. If you wish to limit the search to just yourself, list your own account number.

FIND USERS? A999,B234 (cr)

You may wish to establish different maintenance classes. For example, each class may be able to input only one type of case, or one may be able to add cases using WYLBUR and the other to add cases from magnetic tape. If you intend to establish more than one maintenance class, give each a name and list them (separated by commas).

MAINTENANCE CLASSES? (cr)

Would you like to limit the individuals who can maintain your file? If so, list their account numbers (separated by commas). Otherwise, anyone will be able to add cases to your file.

MAINTENANCE USERS? A999,B234 (cr)

A standard input language has been developed for SPIRES users. The elements making up a case can be arranged in any order. An element may have any number of values of any size. Users who prepare their own data are encouraged to use this flexible language. However, if you already have punched data cards or will receive your data from another installation, you may wish to use a special input language. Please indicate whether you will use the SPIRES, an OTHER, or BOTH input languages.

MAINTENANCE LANGUAGE? spires (cr)

Unless you list account numbers, it is assumed that only the account number you are now using can be used when redefining the file.

DEFINITION USERS? (cr)

You may wish to specify one or more output classes. Output classes can be set up for outputting different sets of elements

and/or for using different output languages. If no output classes are named, then all elements will be given when output is requested. Individuals can limit the elements they receive by setting up their own formats using the display subsystem of SPIRES.

OUTPUT CLASSES? (cr)

A standard output language has been developed for SPIRES that prepares output easily read both by humans and by the SPIRES input language. Users are encouraged to use this flexible language. However, if you wish your data prepared for input to a statistical routine or for magnetic tape output, you may wish to use a special output language. Please indicate whether you will use the SPIRES, an OTHER, or BOTH output languages.

OUTPUT LANGUAGE? spires (cr)

The system requires that each case have a unique means of identification. Whenever cases need to be corrected or deleted, they can be pinpointed by giving the ID number. Unless you specify otherwise, the system will generate a unique ID number and store it with the case every time a new case is added to the file. To indicate that you and not the system will provide the unique identification, answer with the mnemonic of the data element that will contain the identification. (A mnemonic is a string of up to 16 characters that is used to identify values when they are entered into the machine.)

ID? (cr)

You are now ready to define the data elements that can occur in a case. Each data element is specified by giving its mnemonic, followed by a switch mask, followed by a more descriptive name enclosed in quote marks. For example,

DE? T 10000 "Title"

The switch mask indicates whether or not more information has to be supplied for this data element. A zero in the switch mask indicates no and a one indicates yes. Starting from the left the switches have the following meanings:

(1) MINIMUM number of times the data element has to be given a value per case. If the switch is set to zero there is no minimum.

(2) MAXIMUM number of times the data element may be given a value per case. If the switch is set to zero there is no maximum. Data elements that do not take values should have the switch set to one so that the maximum can be set to zero.

(3) VALIDITY of the data element values. If the switch is set to zero a value may be of any length and contain any combination of characters.

(4) ORDER the data element values are stored. If the switch is set to zero values will be stored in the same order they were entered in.

(5) STORAGE of the values. If the switch is set to zero the values will be stored without being converted in any way.

In the example, all the default assumptions may be made except the first--there is a minimum number of occurrences. Consequently, the user will be asked to specify the minimum, as well as what to do when the data element is not given this minimum number of values. Every time a switch is on, the questioning will eventually get around to the problem of what to do when the restrictions are not met. It is far more inexpensive to begin by imposing too many restrictions and removing them later than to attempt to add restrictions after a file has already been built.

What should be done when the data elements in a group are logically related? If the relationship is that the elements occur together, then the group should be treated as a single structural data element. The group is given its own mnemonic and switch settings. The mnemonic is preceded by BEGIN STRUCTURE. After all of the elements in the group have been defined, then END STRUCTURE <mnemonic> is specified. For example,

```
DE?  BEGIN STRUCTURE  D   00000  "DATE"
DE?      MN   01011   "MONTH"
DE?      DY   01011   "DAY"
DE?      YR   01011   "YR"
DE?  END STRUCTURE D
```

Structures may contain other structures as elements. If the group is related in that the occurrence of one of the elements precludes the occurrence of the other elements, then the first of the elements should be preceded by EITHER and each of the rest should be preceded by OR. The elements of an either-or group may be structures. For example,

```
DE?  EITHER ADCD  11101  "ADDRESS CODE"
DE?  OR BEGIN STRUCTURE ADDR  11000  "ADDRESS"
     ....
```

Neither of the two types of data structures should be used unless it is really needed. Both the date and address examples might be better handled by having a single data element that undergoes a

complex validation test and, in the case of the date, undergoes a complex conversion before being stored. Then instead of entering a date value as

```
BEGIN D;  MN=10; DY=2; YR=71; END;
```

it could be entered as

```
D= 10/2/71;
```

Now you will be asked for the data element definitions, one at a time. When there are no more to enter, hit the carriage return. Please enter them in the same order you intend to use when displaying case values.

```
DE?  PAT  11000 "NAME OF PATIENT" (cr)
MIN? 1  (cr)
```

What should be done if the data element is not given a value? If you would like a default value to be supplied you can give a default value in quote marks. If you want a default value to be generated you can give the name and parameter values of one of the system algorithms; type "ALG" and you will receive the algorithms available. If you want to have the whole case discarded do not give a default value.

```
DEFAULT?  (cr)
```

Would you like a diagnostic message put out when the data element is not given any values? If you do not want a message type NO. If you want the message "DATA ELEMENT PAT WAS NOT GIVEN A VALUE" just hit the carriage return. Otherwise type your own message enclosed in quote marks.

```
MESSAGE?  (cr)
MAX? 1  (cr)
```

What should be done if the data element is given more than one value? If you would like the first value saved and the rest discarded type FIRST. If you would like the last value saved and the rest discarded type LAST.

```
DISCARD ALL BUT... FIRST (cr)
```

Would you like a diagnostic message put out when the data element is given too many values? If you do not want a message type NO. If you want the message "DATA ELEMENT PAT WAS GIVEN TOO MANY VALUES" just hit the carriage return. Otherwise type your own message enclosed in quote marks.

```
MESSAGE?  (cr)
```

```
DE?  HRN  11100  "Hospital Record Number"  (cr)
MIN?  1 (cr)
DEFAULT?  (cr)
MESSAGE?  (cr)
MAX?  1 (cr)
DISCARD ALL BUT... FIRST (cr)
MESSAGE?  (cr)
```

There are a number of different ways that a value can be validated. Consequently, we are going to ask for another switch mask with switches for each type of validity test you wish to use. Starting from left to right the switches have the following meanings:

(1) MINIMUM number of characters that may appear in a value. If the switch is set to zero there is a minimum of one.

(2) MAXIMUM number of characters that may appear in a value. If the switch is set to zero there is no maximum.

(3) TYPE of characters that may appear in a value. If the switch is set to zero then any type of character may appear in a value.

(4) LIST or record of permitted values against which the value should be matched. If the switch is set to zero there is no list.

(5) RANGE limits on the value. This option is used only when the value is numeric.

(6) ALGORITHM known to the system that determines whether or not the value is valid.

(7) SPECIAL algorithm the user has created with the help of user consultant.

```
VALIDITY MASK?  0010000 (cr)
```

You may list either the characters that may occur, or, if you precede them with the word NOT, the characters that may not occur. The list should be enclosed in quote marks.

```
CHARACTERS?  "0123456789"  (cr)
```

What would you like to do if the value contains other characters. The alternatives are to STORE the value anyway, or to act as if NO VALUE had been given.

```
IF ILLEGAL?  STORE
```

Would you like a diagnostic message put out when the value is invalid? If you do not want a message type NO. If you want the messages "DATA ELEMENT HRN HAS BEEN GIVEN A VALUE CONTAINING CHARACTERS OTHER THAN 0123456789" just hit the carriage return. Otherwise type your own message enclosed in quote marks.

```
MESSAGE?  (cr)

DE?  BEGIN STRUCTURE DIAG 00000 "DIAGNOSIS"  (cr)
DE?  CD 01000 "CARDIAC DIAGNOSIS"  (cr)
MAX? 1  (cr)
DISCARD ALL BUT... FIRST  (cr)
MESSAGE?  (cr)
DE?  VD 01000 "VASCULAR DIAGNOSIS"  (cr)
MAX? 1  (cr)
DISCARD ALL BUT... FIRST  (cr)
MESSAGE?  (cr)
DE?  AD 11100 "ADMISSION DATE"  (cr)
MIN? 1  (cr)
DEFAULT?  CURRENT DATE  (cr)
MESSAGE?  NO  (cr)
MAX?  1  (cr)
DISCARD ALL BUT... FIRST  (cr)
MESSAGE?  (cr)
VALIDITY MASK?  0000010  (cr)
```

If you would like the list of available algorithms type ALG. Otherwise type the name of the validity algorithm you intend to use.

```
ALG?  DATE  (cr)
DE?  SD 11100  "STATUS AT TIME OF DISCHARGE"
MIN?  1  (cr)
DEFAULT?  "ok"  (cr)
MESSAGE?  NO  (cr)
MAX?  1  (cr)
DISCARD ALL BUT... FIRST  (cr)
MESSAGE?  (cr)
VALIDITY MASK?  0001000  (cr)
```

If the list of values is a data element for some other type of record in the file, then the data element is preceded by RECORD <record name> DE <data element mnem>. Otherwise list the acceptable values separated by commas.

```
VALUE?  CRITICAL, AMBULATORY, DEAD  (cr)

DE?  END STRUCTURE DIAG  (cr)
DE?  (cr)
```

Would you be willing to go back through the data elements predicting how often each will occur and approximately how long

each of the values will be? From this we could estimate the space needed for your file more accurately. Please answer YES or NO.

SIZE AND OCCURRENCE PREDICTIONS? NO (cr)

Since you are not willing to predict, are you willing to have statistics collected concerning this information so that when you redefine the file you can answer the question? answer YES or NO.

SIZE AND OCCURRENCE STATISTICS? YES (cr)

Thank you for defining your file. When the computer prints DEFINITION OK the file will be created. Otherwise the definition processor will ask you some more questions.

DEFINITION OK. YOU WILL BE SENT BACK TO WYLBUR.

****

CHAPTER 5:
FILE MAINTENANCE AND UPDATE

5.1 Concepts and Definitions:

5.1.1 General

After the file manager has defined his file (see chapter 4), he must have the means of maintaining it. In particular, he must be able to add, replace, delete, or modify records and data elements in his file. File maintenance facilities are provided by the File Management Command Language and the WYLBUR Text-Editing Language. The use of these two languages allows the file manager to perform record or data element modifications to his file. The file manager also has available two methods of updating his file. He can key at the terminal the various modifications he desires, or, he can operate in batch mode using the batch update program (see 5.4).

5.1.2 External Data Format

Data records exist within the SPIRES system in one or more of three possible formats:

(1) external
(2) intermediate
(3) internal

The existence of the last two formats is hidden from the user, who interacts exclusively with the external format. As mentioned in chapter 4, the user has the option of using the standard SPIRES external format. (It is hoped that implementation experience will lead to more flexibility in this format for greater user convenience.) The following discussion will center on the SPIRES standard external format.

5.1.2.1 Data Record Delimiters

The commands BEGIN;...END; define the limits of one record. All data elements specified within the limits of BEGIN;...END; constitute one logical record (or some portion of one) in a user file.

5.1.2.2 Specification of Data Element Values

There are three kinds of Data Element Value commands. In all cases, the equals sign is optional. In case (1), if the equals sign is omitted a blank must be substituted.

(1) <MNEMONIC> = <VALUE>

This command should be used in cases where the value contains no embedded quotes or semicolons.

(2) <MNEMONIC> = "<VALUE>"

This command should be used whenever the value contains leading blanks or one or more semicolons or double quotes. Embedded double quotes (") are represented as (""); e.g., "the title is ""Quark Models."""" Alternatively, one may use single quotes as delimiters where double quotes are contained within the value; e.g., 'the title is "Quark Models."'

(3) <MNEMONIC>

This command is termed the NULL command. It is used whenever the user finds it sufficient to merely indicate the existence of a data element, without specifying a value. It could also be used to indicate "on" for a data element with only two possible values.

5.1.2.3 Specification of Data Element Structures

All data element values with a hierarchical relationship to some other data element are surrounded by the delimiting commands

BEGIN <structure name>;...END;

These may be nested one within another; the user, when inputting, is encouraged to use indentation to help him keep track of the relationships. The system, when displaying a record for modification, will indent structures automatically.

5.1.2.4 External Format Example

```
BEGIN;
A="Smith, Roger J.";
PAG=256p;
PREPRINT;
T=QUARK MODELS;
BEGIN DATE;
     MO=07;
     DA=21;
     YR=70;
END;
END;
```

5.2 On-line Entry & Deferred Update

On-line entry refers to the collection, editing, syntax checking, and partial validity checking of a file update entered at the terminal; deferred update refers to the queuing of a file update for a batch update program that is scheduled for execution at some later time.

In order to execute a deferred update in SPIRES, the file manager uses the following command sequence:

(1) A SPIRES update command is issued to add, substitute transfer, or remove records in the file.

(2) If the record is to be modified, it is listed in the user's WYLBUR working data set (WDS) in external form (see 5.1.2). When a record is to be added or substituted, nothing is written in the user's working data set.

(3) At this point the user collects and/or edits the record or data elements by using WYLBUR text-editing commands.

(4) The user completes the update by keying "UPDATE" to SPIRES.

(5) SPIRES now reads the information contained in the user's WDS and transfers the record information, now in intermediate format, into the deferred update queue.

(6) If a new record is to be added or substituted for an existing record, the user is placed in WYLBUR collect mode. After the collecting for a record is complete, the user proceeds as in (3) to (5) above.

(7) If an existing record is to be removed, the user specifies its key element value (see below) in conjunction with a REMOVE command.

5.3 On-line File Management Commands

On-line file management commands make use of a key data element whose simple value is the unique identification of the record being processed. A record ID number is one example of a key data element.

5.3.1 Update Execution Commands

(1) COMMAND: UPDATE

PURPOSE: To complete a pending TRANSFER, SUBSTITUTE, or ADD command (see 5.3.2 below) by applying the contents of the user's WDS to a file.

ACTION PERFORMED: Generates a record from the WDS and passes the record to the batch queue for later processing by the batch update program. Additional information is supplied indicating TRANSFER, SUBSTITUTE, or ADD and which key element value to use. Subsequent attempts to retrieve the record will yield the updated version from the batch queue.

WHEN VALID: In response to "-?" with pending TRANSFER, SUBSTITUTE, or ADD.

CANCELLATION: Self-cancelling. See DEQUEUE Command.

DIAGNOSTICS: RECORD XXX exists.
RECORD XXX does not exist.
RECORD XXX data element XXX.nnn does not exist.
INVALID STRUCTURE.
RECORD NOT SPECIFIED.

EXAMPLE: UPDATE 576

(2) COMMAND: DEQUEUE <key element value>

PURPOSE: To remove from the batch update queue the cumulative updates that have been completed during the day against the record indicated.

ACTION PERFORMED: Purges the changed data base record image from the batch update queue, and deletes the queue pointer from the data base record.

WHEN VALID: In response to "-?"

CANCELLATION: Not applicable.

DIAGNOSTICS: NO UPDATES QUEUED FOR RECORD=XXX.
MISSING OPERAND--ID OR ALL NOT SPECIFIED.

EXAMPLE: DEQUEUE 576

5.3.2 Record and Data Element Commands:

(3) COMMAND: TRANSFER <key element value>

PURPOSE: Allows the user to update an existing record by applying the WYLBUR text-editing commands to the specified record.

ACTION PERFORMED: The command causes SPIRES to retrieve and write into the user's WDS the specified record in external format. This replaces any previous contents of the WDS. The lines brought into the WDS by this command are sequentially numbered starting with 1. If the record is currently being processed by another user, a message is given to the would-be user indicating conflict, and the TRANSFER command is ignored. The record information is automatically printed out at the user's terminal. The user can terminate this printing by depressing the ATTN key.

WHEN VALID: In response to "-?"

CANCELLATION: Another file management command or the LOGOFF command.

DIAGNOSTICS: SYNTAX ERROR
OPERAND MISSING
RECORD BUSY
NO SUCH RECORD EXISTS

EXAMPLE: TRANSFER 568

(4) COMMAND: ADD

PURPOSE: To add one new record.

ACTION PERFORMED: The command causes SPIRES to place the user in collect mode. Previous WDS contents are cleared unless NOCLEAR is specified. Collection begins with the first allowable line number, usually 1.

WHEN VALID: In response to "-?"

CANCELLATION: Another file management command or the LOGOFF command.

DIAGNOSTICS: SYNTAX ERROR

SPECIAL NOTE: If the key element value is not generated by the system, it must be collected by the user along with the other data elements making up the record.

EXAMPLES: ADD NOCLEAR
ADD

(5) COMMAND: SUBSTITUTE <key element value> (CLEAR or NOCLEAR)

PURPOSE: To replace an existing record with a new one.

ACTION PERFORMED: The command causes SPIRES to place the user in collect mode. If the old record is currently being processed by another user, a message is given to the would-be user indicating conflict, and the SUBSTITUTE command is ignored. If "CLEAR" or "NOCLEAR" is not specified, the prompt "OK TO CLEAR?" occurs, with Y, YES, and carriage return acceptable affirmative responses and N and NO acceptable negative responses. Collection begins with the first allowable line number, unsually 1.

WHEN VALID: In response to "-?"

CANCELLATION: Another file management command or the LOGOFF command.

DIAGNOSTICS: SYNTAX ERROR
OPERAND MISSING
RECORD BUSY
NO SUCH RECORD EXISTS

EXAMPLE: SUBSTITUTE 578

(6) COMMAND: REMOVE <key element value>

PURPOSE: To remove an entire record from a user file.

ACTION PERFORMED: Places REMOVE command in deferred update queue (and logically removes the record from the file).

WHEN VALID: In response to "-?"

CANCELLATION: self-cancelling.

DIAGNOSTICS: SYNTAX ERROR
OPERAND MISSING
NO SUCH RECORD EXISTS

EXAMPLE: REMOVE 578

(7) COMMAND: DISPLAY <key element value> (IN WYLBUR)

PURPOSE: To fetch for display an existing record. No previous search request is necessary.

ACTION PERFORMED: The command causes SPIRES to retrieve the specified record and print it out at the user's terminal. The user can terminate the printing by depressing the ATTN key. If the option "IN WYLBUR" is included, the record information is automatically written into the user's WDS in external format. This replaces any previous contents of the WDS. The lines brought into the WDS by this command are sequentially numbered starting with 1.

WHEN VALID: In response to "-?"

CANCELLATION: Self-cancelling.

DIAGNOSTICS: SYNTAX ERROR
OPERAND MISSING
NO SUCH RECORD EXISTS

EXAMPLE: DISPLAY 578

5.4 Batch Services:

5.4.1 Batch Update

Batch entry/batch update is provided for users who do not wish to enter updates on-line. The user must provide a sequential card-image data set, consisting of one or more records in external form (see 5.1.2). In addition to providing BEGIN;...END; groups, the user must indicate the operation to be performed--ADD, SUBSTITUTE, or REMOVE. One of these three commands is required to precede the command.

Example: A Batch Update Data Set

```
ADD;
A= ...
T= ...
     .
     .
     .
END;
REMOVE 12345
SUBSTITUTE 12896
A= ...
T= ...
     .
     .
     .
END;
ADD;
     .
     .
     .
  etc.
```

An additional feature is offered to the batch update user. He may specify a set of default values to be applied should certain data element values not be specified in a particular ADD record. The default values specified remain in effect during the update run until another default definition appears in the input, or until an explicit "CANCEL DEFAULT" is encountered.

Example:

```
DEFAULT;
DATE="7 Jan 1970";
AFFIL=SLAC;
PREPRINT;
END;
ADD BEGIN;
T="QUARK MODELS";
A="SMEDLY,J.R.";
SOURCE="AMER.JOUR.PHYS";
END;
  .
  .
  .
CANCEL DEFAULT;
```

5.4.2 Batch Conversions

It is expected that many users will wish to build files using machine-readable data acquired from government agencies, professional societies, etc. Examples are "NUCLEAR SCIENCE ABSTRACTS," "DESY HEP FILE," "AIP SPIN," "Library of Congress MARC," etc. Programming services will be provided for the design and implementation of routines to translate from the original format into SPIRES intermediate format. Costs will be borne by the user, except in those instances where SPIRES I conversion programs now exist and are being used.

5.4.3 Batch Utilities

The following is an initial list describing the utility programs necessary to administrate, control, and recover the overall collection of SPIRES/BALLOTS files. In general, the user will be unaware of the existence of these utilities.

5.4.3.1 VALIDATE: Enough redundancy has been built into the record and file structure design to permit programmed validation of files without frequent recourse to manual checking. It is felt that during subprocessor debugging such a program would be invaluable. During initial production, the program could be run once daily on files having heavy update activity, and less often on other files. The use of this program would diminish as more confidence is gained in the file services portion of the subprocessors. It is essential that each file problem be discovered as early as possible, before its effect is compounded by the simultaneous occurence of other problems. The time the problem occurred must be precisely known, so that the problem can be compared with changes in environment.

5.4.3.2 DBIDXLST: This program will provide a formatted listing capability for both index and primary data records (see 4.2). The user of the program may submit as input a range, a list, or some other combination to specify those records he wishes to see.

5.4.3.3 FULRES: This program will restore block images from tape to disk. The disk file thus created will be a precise image of the file from which the tape was created.

5.4.3.4 FULDUMP: This program is the converse of the one immediately above. It copies the precise images of the blocks in a specified file from an STSM data set to tape.

5.4.3.5 DEFUPDT: The deferred update program will, given an input stream of updates keyed in during the day, make appropriate insertions, deletions, or modifications to file records in internal formats. The input will come from the batch build program, from the on-line deferred update queue, and from batch conversion programs (e.g., MARC, DESY, NSA, etc.).

5.4.3.6 IDXREBLD: The index rebuild program will build a new index from an existing data base should the old index be destroyed or rendered useless in some way.

5.4.3.7 PARDUMP: The partial dump program will dump all blocks modified in a deferred update run.

5.4.3.8 PARES: The partial restore program will restore selected records previously copied to tape by the PARDUMP program.

5.4.3.9 DISKMAP: This program will create a listing that shows, by disk pack, which files or file extents are occupying space, and where there are empty spaces. It will also show where extent locations are by file set. The purpose of such a program will be to allow the user consultant to allocate space for new files intelligently. It will also tell him where a fragmented file set requires consolidation to increase the size of available space chunks.

5.4.3.10 LOGREC: This program will reconstruct the batch queue from the log tape for the day (or for previous days if necessary).

5.4.3.11 LOGSTAT: This program will do a reduction on the log tape for the day to produce command usage statistics, user error statistics, and summaries of user activity.

5.4.3.12 DSALLOC: This program will request an initial allocation for a file set, or request an additional extent for a file that is almost out of space. It will also update the DSDIR (data set directory), which is the SPIRES/BALLOTS analogue of the STSM file directory.

5.4.3.13 DSDIREC: This program will reconstruct DSDIR should it be lost or damaged, by using the STSM file directory. (DSDIR, as mentioned above, contains information that also exists in the STSM directory.) The purpose of DSDIR is to permit the use of STSM files from the batch partition. DSDIR is the first series of records on the SPIRES/BALLOTS packs, and it is thus only necessary to open the SPIRES/BALLOTS packs as a multivolume O/S data set in order to get to DSDIR and find where the other logical data sets reside.

5.4.3.14 AVSPREC: For each file, there will exist one block that contains pointers to available space chains within the file. This program reconstructs the chains should one or more be broken during a system failure.

****

CHAPTER 6: SEARCHING AND RETRIEVING

6.1 General

Retrieving information from a file consists of two operations: locating relevant material (called "searching," i.e., finding the record's physical location); and presenting that material (called "display," i.e., displaying the results of the search on some output device). The retrieval process is carried out in several steps that may be repeated until the user is satisfied that he has obtained all the information from the file he needs at the time. First, he must formulate a request that specifies the properties of the information he wishes to retrieve. Examples of these properties are (1) the name of an author, (2) a patient record number, (3) a specific date, (4) a range of ages, (5) a set of items from an index file.

The user types his request into the system via his terminal. The system will acknowledge the request and initiate a search. Upon completion of a search operation, the system will respond with the number of records that meet the criteria given in the request. These records constitute a subset of the file called the "search results." It may contain a very large number of records or none at all.

At this point, the user may wish to look at the contents of one or more of the records. To accomplish this, he must type in an output request. The system will respond by displaying the desired information at his terminal. The user may interrupt the display at any time, by depressing the ATTN key, and enter a different request.

After receiving the response giving the number of found records or after viewing some or all of those records, the user may enter a request specifying additional criteria. The system will execute another search that may increase or decrease the number of records in the search results. In this manner, the user may refine his search until he has derived a subset of the file that contains, hopefully, all the relevant material and no irrelevant material. He may then request that this set be listed on a high-speed printer, be moved to a WYLBUR working data set, or be displayed at his terminal.

The following sections give detailed descriptions of each type of request that may be entered by users and also some instruction in the efficient use of the system.

6.2 Search Language

The commands available to a user in the search function are grouped in two classes. The first is the set of commands used to browse indexes and the second is the set of commands used to search the chosen file.

6.2.1 BROWSE COMMANDS

If the user is unaware of the contents of a particular index, he may explore for terms of interest to him via the BROWSE commands. He may browse the entire index or only a part of it, depending on the specificity of his prior knowledge. BROWSE commands are usually issued iteratively, with each succeeding command restricting further the scope of index terms to be displayed.

6.2.1.1 The BROWSE <index name> INDEX (integer) Command

This form of the browse command searches the entire index. The user receives back "n" index terms evenly distributed across the collating sequence, where n is either the integer specified in the command or a default of 10. Shown with each term is the number of data records it indexes.

Example: BROWSE topic INDEX

	TOPIC INDEX	NUMBER
1.	asparagus	10
2.	cabbage	25
3.	endive	2
4.	green beans	12
5.	kohlrabi	4
6.	mint	15
7.	peach	20
8.	summer squash	6
9.	tangerine	16
10.	wheat germ	15

6.2.1.2 The BROWSE <value> - <value> (integer) Command

After issuing the BROWSE...INDEX command, the user may drop to an intermediate level. The system will display a list of index terms within the inclusive range specified and the number of data records containing each term. The fact that more undisplayed terms exist between two entries in the list is indicated with an ellipsis (...). It should be noted that in those instances where an inordinate amount of disk access would

be required to locate n terms, some significant number less than n will be displayed. This limitation may be circumvented by subsequently issuing more refined BROWSE commands. (Once the window size--integer--has been specified, another specification is not needed unless the user wishes to change it.)

Example: BROWSE cantaloupe - eggplant

TOPIC INDEX	NUMBER
1. Cantaloupe ...	3
2. Celery ...	64
3. Corn	2
4. Cucumber	9
5. Dates ...	16
6. eggplant ...	11

then: BROWSE cantaloupe - celery

TOPIC INDEX	NUMBER
1. Cantaloupe	3
2. Carrots ...	21
3. Celery ...	64

6.2.1.3 The BROWSE <right-truncated value> (integer) Command

This command implicitly defines a range. For example, if SEC# were specified as a truncated value, a distribution of n or less index terms would be displayed within the interval SEC--up to, but not including, SED.

Example: BROWSE Ca#

TOPIC INDEX	NUMBER
1. Cantaloupe	3
2. Carrots	21
3. Catnip ...	3

6.2.1.4 The BROWSE <value> (integer) Command

This command returns to the user a list of n or less index terms with the specified value centrally located in the list. Previous comments regarding indications of undisplayed terms and restrictions on the number of disk accesses apply.

Example: BROWSE celery

TOPIC INDEX	NUMBER
1. Cantaloupe	3
2. Carrots	21

3. Catnip 3
4. Celery 64
5. Corn 2
6. Cucumber ... 9

6.2.2 Retrieval Commands:

6.2.2.1 Record Selection Commands

(1) The FIND Command

FIND <index name> <index value> (SYNONYMS)
or FIND <data element name> <data element value>

If a user has selected a file, knows the names of some or all of the indexes, and has one or more values in mind, he may formulate and enter a search request.

Examples: a. -? FIND AUTHOR smith
17 FOUND

b. -? FIND Patient "Ronald C. White"
18 FOUND

A search request is composed of the name of at least one index and at least one value for that index, or of at least one data element and at least one value for that data element. It should be noted, however, that the FIND <data element name> command requires a sequential scan of the data base when that element is not indexed, which is very inefficient. It will, therefore, be necessary to restrict this command to files of relatively small size or to small subsets of larger files. If more than one index-value pair is contained in a request, each pair must be connected with one of the logical operators "and," "or," "not." A request containing only one pair is called a simple request. A request containing more than one pair is called a compound request.

```
Examples: a. (simple)    -?  FIND TITLE honor
          b. (simple)    -?  FIND TOPIC nobility
          c. (compound)  -?  FIND AUTHOR jones
                         -?  AND TITLE magnet
                    (or)-?  FIND AUTHOR jones AND TITLE magnet
          d. (compound)  -?  FIND AUTHOR tom
                         -?  AND AUTHOR dick
                         -?  AND AUTHOR harry
                    (or)-?  FIND AUTHOR tom dick harry
          e. (compound)  -?  FIND AUTHOR HUNTER
                         -?  AND TITLE pray
                         -?  OR TITLE reward
```

```
(or)-?  FIND AUTHOR hunter AND TITLE pray
    -?  OR TITLE reward
```

In example c, the system would find all works by someone named Jones with the word magnet in the title. In example d, any work by Tom, Dick, and Harry would be retrieved. In example e, all works by Hunter containing the word pray in the title, or all works containing the word reward in the title, would be found. If it was intended that the works with reward in the title must also have been by Hunter, parentheses could be used in the request. The precedence of logical operators is left to right, with grouping capabilities provided by the use of parentheses. A new search is started by specifying the command verb "FIND."

Example: FIND AUTHOR hunter AND (TITLE pray OR TITLE reward)

If several values are specified for one index in a request, the name of the index need not be repeated.

Example: FIND TITLE happy OR gleeful OR ecstatic

When the system responds to a FIND request with a count of the number of records that match the specified criteria, the count may fall into any of three categories. In the first case, the number found will be about what the user expected and he will wish to see the contents of some or all of the records. See chapter 7 for a description of output commands.

In the other two cases, the number is too small or too large. If it is too small, a request may be entered that will increase the number of records found.

```
Example:    -?  FIND AUTHOR smith
                1 FOUND
            -?  OR AUTHOR jones
                4 FOUND
```

If the number of records found is too large, the user may enter a request that will reduce the set of records.

```
Example:    -?  FIND TITLE history
                186 FOUND
            -?  AND TITLE england
                14 FOUND
```

If a user has defined a file such that one or more indexes have a synonym capability, then he may use the FIND...SYNONYM option. In this case, the index term given in the FIND command argument is combined logically with all synonyms for the term in that index. The results are those for the given index term plus those for the synonyms.

Example: -? FIND TOPIC Civil War SYNONYM
ALSO War Between States

75 FOUND

(2) The Arithmetic Commands

FIND <index> op <value>
or FIND <data element> op <data element value>

where op is "<," ">," or "=."

Examples: FIND Age > 55
FIND Number < 765

(3) The DIFFERENCE Command

DIFF <index1 - index2> op <value>
or DIFF <data element1 - data element2>
op <value>

where op is "<," ">," or "=."

Example: DIFF (age - RxDuration) < 12

(4) The DATE Qualifier to
a FIND Command

AND DATE (AFTER/BEFORE/SINCE) <date>
or: AND DATE BETWEEN <date> AND <date>
or: AND DATE FROM <date> to <date>

A user may wish to limit the records he retrieves to some span of time, such as documents published after 1967. If he is using a file with a date qualifier to the index, he may add a qualifying phrase to his search. A date phrase may not be entered as the only term in a search request.

Examples: a.
```
-? FIND TITLE subsonic
   12 FOUND
-? AND DATE AFTER 1968
   8 FOUND
-? TYPE
```

b.
```
-? FIND AUTHOR rubin
   AND DATE BETWEEN jun, 67 AND aug, 69
   4 FOUND
-? TYPE
```

c. -? FIND TITLE pastel AND AUTHOR minton
AND DATE BEFORE Sep, 68
1 FOUND
-? AND DATE BEFORE Mar, 69
3 FOUND
-? TYPE

d. -? FIND DATE AFTER 1960
CANNOT ENTER DATE ALONE

6.2.2.2 The EXTRACT Command

EXTRACT <data element(s)>
EXTRACT WYLBUR <data element(s)>
EXTRACT COMPUTE <data elements(s)>

Example: EXTRACT Name

Data element "Name" will be extracted from the file and all values obtained (extracted) will be printed out at the user's terminal.

Example: EXTRACT WYLBUR Name

Data element "Name" will be extracted from the file and placed in the user's WYLBUR DATA SET.

6.2.2.3 The CROSS Command

This command allows the user to determine all the possible combinations of entries between two fields. A table is printed out giving the results.

Example: CROSS 25 WITH 38

For instance, if field 25 can take on the values Blank, Y, or N, and field 38 takes the values Blank, A, B, C, then the following table might be generated:

		38				
		--	A	B	C	
	--	13	40	25	10	88
25	Y	10	65	78	32	185
	N	8	42	69	70	189
		51	147	172	112	462

6.2.2.4 The STORE Commands

(1) STORE FILE <filename>

One form of the STORE command allows the user to save a specified subset of his file. This is particularly helpful to the user since he then does not have to re-select a particular subset each time a new search strategy is begun. This command sets up a completely separate STSM file including all the parameters of the mother file, except, of course, the STSM file name and security key. The actual creation of the subset file is accomplished in batch mode during the next overnight service block.

Example: STORE FILE myfile
SECURITY KEY? 1526 (cr)

If the user executes this particular STORE command he then has the ability to logon and retrieve this file by using the filename in a SELECT file command.

(2) STORE RESULT <resultname>

Another form of the STORE command allows the user to store the results of a specified search strategy. This allows the user to save results intermittently during the same session and then at a later time recall those results.

Example: STORE RESULT hisfile

Note that the STORE command is used in conjunction with the SELECT command. In other words, when the user wishes to retrieve a file (or a set of record pointers) he uses the following SELECT commands for STORE FILE command and the STORE RESULT command, respectively.

Example: SELECT FILE myfile
or: SELECT RESULT hisfile

6.2.2.5 The BACKUP Command

A user may go through many search iterations, entering successive requests to reduce or expand the set of records that meet his specifications. Occasionally, he may enter a request that reduces or expands the search result by much more than is desirable at that point. Rather than have to re-enter his entire search
specifications, he may then use the BACKUP command. This command allows the user to request that his search status be reset to what it was before the last request. If a user enters a request that reduces the set of records found to zero, the system automatically backs up to its status before that request.

Examples: a. -? FIND TOPIC environment
327 FOUND

```
-?  AND TOPIC pollution
    83  FOUND
-?  AND TOPIC water
    17  FOUND
-?  AND AUTHOR bentley
    1   FOUND
-?  BACKUP
    RESULTS RESET TO 17
-?  AND (AUTHOR bentley OR judson
    OR farnum)
    5 FOUND
```

b.

```
-?  FIND TITLE merchant
    14 FOUND
-?  AND TITLE vanice
    0 FOUND; RESULTS RESET TO 14
-?  AND TITLE venice
    1 FOUND
```

c.

```
-?  FIND TOPIC drugs
    228 FOUND
-?  AND TOPIC psychosis
    93 FOUND
-?  AND AUTHOR danzig
    12 FOUND
-?  AND TOPIC therapy
    1 FOUND
    BACKUP
    RESULTS RESET TO 12
-?  AND (AUTHOR danzig AND
    fridley)
    7 FOUND
```

6.2.2.6 CLEAR RESULT <resultname> Command

This command will cause a previously stored set of search results to be purged.

Example:

```
-?  SELECT FILE BOOKLIST
-?  FIND AUTHOR Black
    22 FOUND
-?  AND (TITLE fuliginous OR melanism)
    11 FOUND
-?  STORE RESULT MYFIND
-?  LOGOFF
      .
      .
      .
    (new session)
-?  SELECT RESULT MYFIND
    11 FOUND
```

```
-?  AND DATE AFTER 1969
    4 FOUND
-?  LIST OFFLINE
      .
      .
      .
-?  CLEAR RESULT MYFIND
```

6.2.3 Batch Searching

The user may elect to enter a search command into the system for a particular file, and have all additions to the file passed against that command periodically. In order to do this, he collects a single or multi-line search command set in a WYLBUR working data set and, after making appropriate corrections, enters the command "BATCH."

Example:

```
    -?  COLLECT
1.  -?  SELECT FILE ECONOMICS
2.  -?  FIND AUTHOR Samuelson
3.  -?  AND TOPIC GNP
4.  -?  *** (ATTN)
    -?  BATCH
```

The user will now receive a weekly printout listing all the new entries in the file that fulfill the stated criteria.

****

CHAPTER 7: DISPLAYING OUTPUT

7.1 Output Formats:

7.1.1 System Default Format

If no format is designated by the user for output, SPIRES will use its own format in default.

7.1.2 Formats Defined at File Definition Time:

7.1.2.1 SELECT FORMAT <format name> Command

A display format determines the set of elements to be displayed for each record and the order in which they are presented. Each file has at least one format (the default format) defined for it. At any time, the user can select any of the various formats defined by entering the SELECT FORMAT command. All his displays will then be governed by the selected format until he selects another one. If the user does not specifically select a format, the system will automatically select the standard format defined for that file.

```
Example:    -?    SELECT FILE economics
                  FILE READY
            -?    FIND AUTHOR  house
                  2 FOUND
            -?    TYPE
                  1.  TITLE   tax shelters
                      AUTHOR  i.r. house
                      DATE    1933
                  2.  TITLE   capital gains in real estate
                      AUTHOR  i.r. house
                      DATE    1964
            -?    SELECT FORMAT  pub
            -?    TYPE
                  1.  AUTHOR  i.r. house
                      TITLE   tax shelters
                      PUB     houghtin mifflin
                      DATE    1933
                  2.  AUTHOR  i.r. house
                      TITLE   capital gains in real estate
                      PUB     prentice hall
                      DATE    1964
```

7.1.3 Formats Defined During Session Time:

7.1.3.1 DEFINE FORMAT <format name> Command

If none of the formats defined for a file meet a user's needs in a particular session, he can use this command to create a new format. The system will prompt him successively for the name of each element he wishes to include in the format. The elements will be displayed in the same order in which he enters their names. When all names have been entered, the user answers the prompt by a carriage return. The newly defined format will be selected automatically, and will remain in effect until a SELECT FORMAT command is issued or the user issues the LOGOFF command.

```
Example:    -?    FIND TITLE  desert
                  2 FOUND
            -?    DEFINE FORMAT  form
                  LIST ELEMENTS
                  1.
            -?        DATE
                  2.
            -?        TITLE
                  3.
            -?        AUTHOR
                  4.
            -?        CITY
                  5.
            -?    (cr)
            -?    TYPE

                  1.  DATE     Sep 16, 1965
                      TITLE    aridity of the ghobi desert
                      AUTHOR   i.m. dryer
                      CITY     philadelphia
                  2.  DATE     Apr, 1968
                      TITLE    sands of the sahara desert
                      AUTHOR   n.o. waters
                      CITY     denver
```

7.2 Output Language:

7.2.1 The TYPE (SELECT) (range of rec #'s) (Data Element List) (UNLABELED) Command

This command will cause the system to begin presenting the data elements listed in records found in the previous search request directly at the user's terminal. The records are presumed to be ordered in some way specified by the file manager; for instance, by publication date. The range of numbers specifies the first and last record to be displayed. For instance, in the command "TYPE 4/17," the first record to be

displayed will be record 4, the next will be record 5, and so on until record 17 has been displayed. If no range is given, the display will start with the first record and continue through the last one. The user may stop the display at any point by depressing the ATTN key. The system will terminate the display with the last character printed before "ATTN" was entered and type out the prompt "NEXT:" If the user responds to this prompt with the word YES or a carriage return, the system will resume the display at the beginning of the next record. If the user responds with the word NO or by pressing the ATTN key, the display will be considered complete and the "-?" prompt will be typed. The user may then enter any request he wishes. If "UNLABELED" is specified, the data element or index names are not printed. Otherwise, labels are printed by default.

```
Example:    -?    TYPE 3/5
                  3. TITLE    mandarin history
                     AUTHOR   pierre l'orange
                     DATE     july 11, 1966
                  4. TITLE    the red planet in ... (attn key)
                  NEXT:
              -?  YES
                  5. TITLE    the neutral mouse
                     AUTHOR   grey, smokey
                     DATE     1957
              -?
```

The SELECT option will cause the user to be interrogated following the typing of each record with the prompt "KEEP?". If the response is "YES" or a carriage return, the record is written into the WYLBUR WDS. If any other response is given, the record is not written into the WDS. After the response is received, the system proceeds to display the next record.

7.2.2 The OUTPUT <media> Command

```
<media> = WYLBUR <options>
          OFFLINE <options><priority><bin #>

<options> = <list of entry numbers> and/or
            SORT <sort options>

<sort options> = ASCENDING BY <data element list>
                 DESCENDING BY <data element list>

<priority> = IDLE
             OVERNIGHT
             STANDARD
             EXPRESS
             URGENT
```

Examples: a. -? OUTPUT OFFLINE 3,6-22 SORT ASCENDING
BY AUTHOR PRIORITY URGENT BIN 439

b. -? OUTPUT WYLBUR SORT DESCENDING BY DATE

c. -? SELECT FORMAT MYFORMAT
-? OUTPUT WYLBUR

The OUTPUT WYLBUR command will write the search results into the user's WYLBUR working data set in the specified sort order, using whichever output format the user selected at the time of the OUTPUT command.

The OUTPUT OFFLINE command will list the records found in the search on a printer located at the computer site rather than on the user's terminal. The reason for using this command is to avoid having a large set of records listed at the terminal, which is much slower than the printer. The user may request that the printing be done overnight, for which he will be charged less.

7.2.3 CRT Commands

An appropriate set of commands will be implemented to accomplish paging, scrolling, and the display of information on a scope face (see 3.2.2).

END

DEPT.OF HEALTH
EDUCATION AND
WELFARE

U.S. OFFICE OF
EDUCATION
ERIC
DATE FILMED

6 21 71